

# Spark Streaming

Marcos, Jeferson, Jonatas

# SUMÁRIO



INTRODUÇÃO

SPARK STREAMING

Para que serve? Para que não serve?

ARQUITETURA

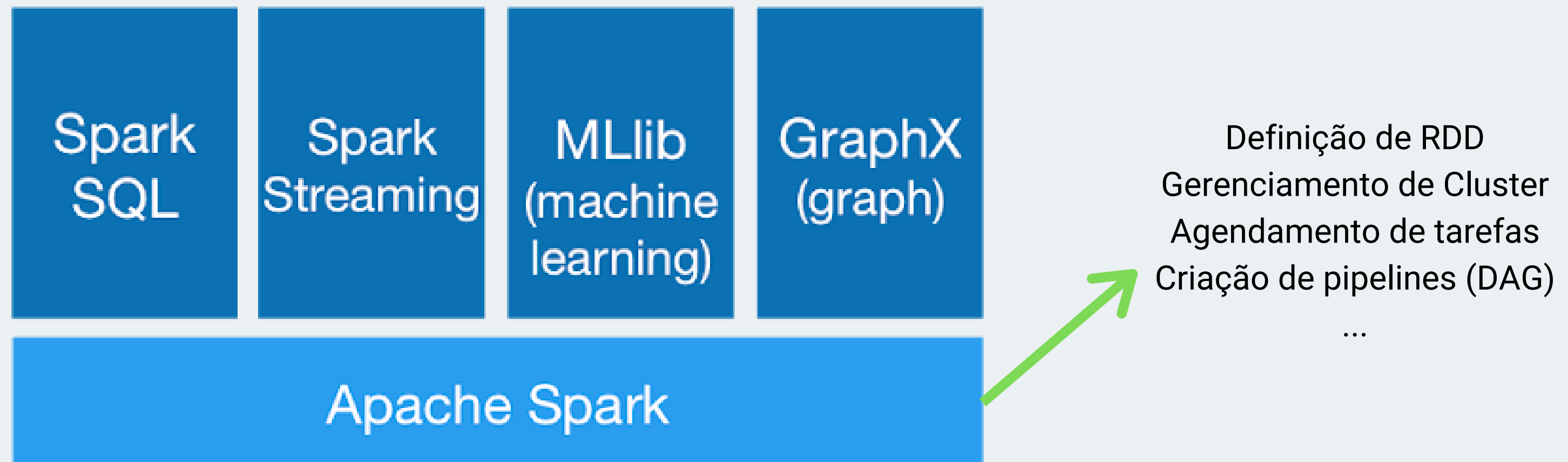
INSTALAÇÃO

APLICAÇÃO

Estatísticas do Twitter em tempo real

# INTRODUÇÃO SPARK

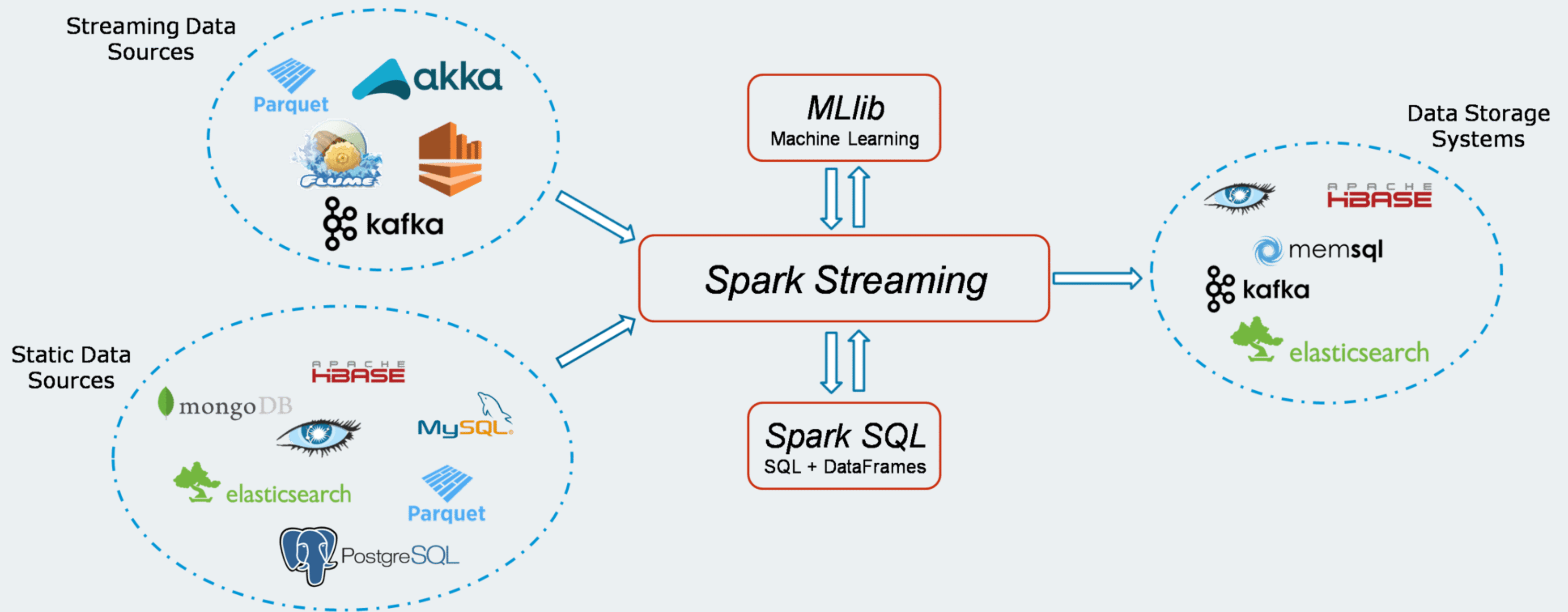
O Apache Spark é um middleware para processamento de dados projetado para ser **rápido** e **acessível**.



*Figura: Stack do Spark*

# SPARK STREAMING

O Spark Streaming permite o processamento de streaming de dados em tempo real.



# SPARK STREAMING

Para que serve? Para que não serve?

## SERVE

- *"Low latency processing applications" (0.5-2 segundos)*
- Trending-topics em redes sociais
- Modelagem dos visitantes de um site
- Monitoramento de logs

"Linear scaling to 100 nodes, sub-second latency, and sub-second fault recover"

## NÃO SERVE

- Latência abaixo de algumas centenas de milissegundos
- Processamento em batch
- Trading de alta frequência

# SPARK STREAMING

Para que serve? Para que não serve?

## SERVE

- *"Low latency processing applications" (0.5-2 segundos)*
- Trending-topics em redes sociais
- Modelagem dos visitantes de um site
- Monitoramento de logs

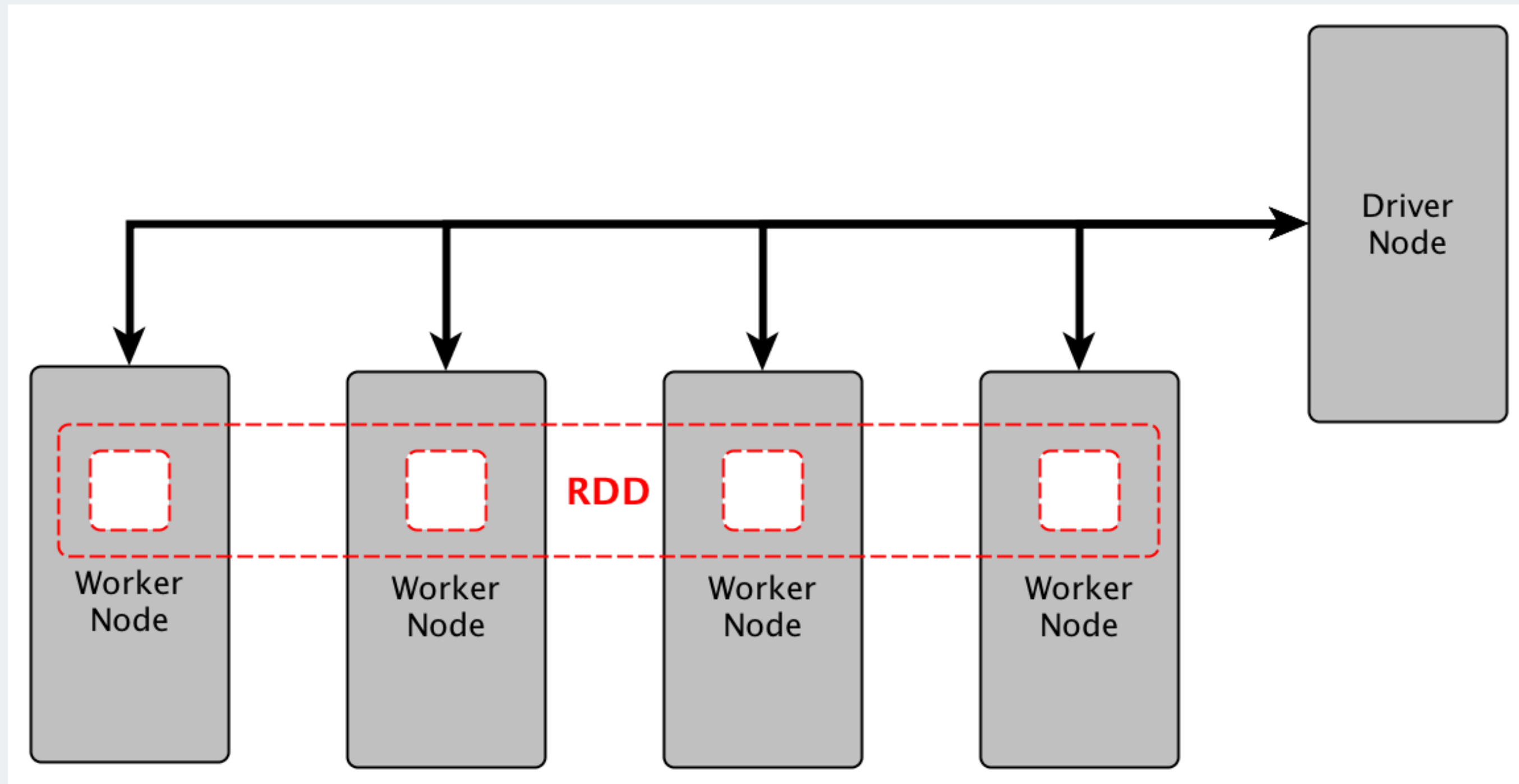
## NÃO SERVE

- Latência abaixo de algumas centenas de milissegundos
- Processamento em batch
- Trading de alta frequência

The Netflix logo, consisting of the word "NETFLIX" in a bold, white, sans-serif font with a slight 3D effect, set against a solid red rectangular background.

- **video insights**
- **ML pipeline for recommendations**

# Arquitetura - RDD

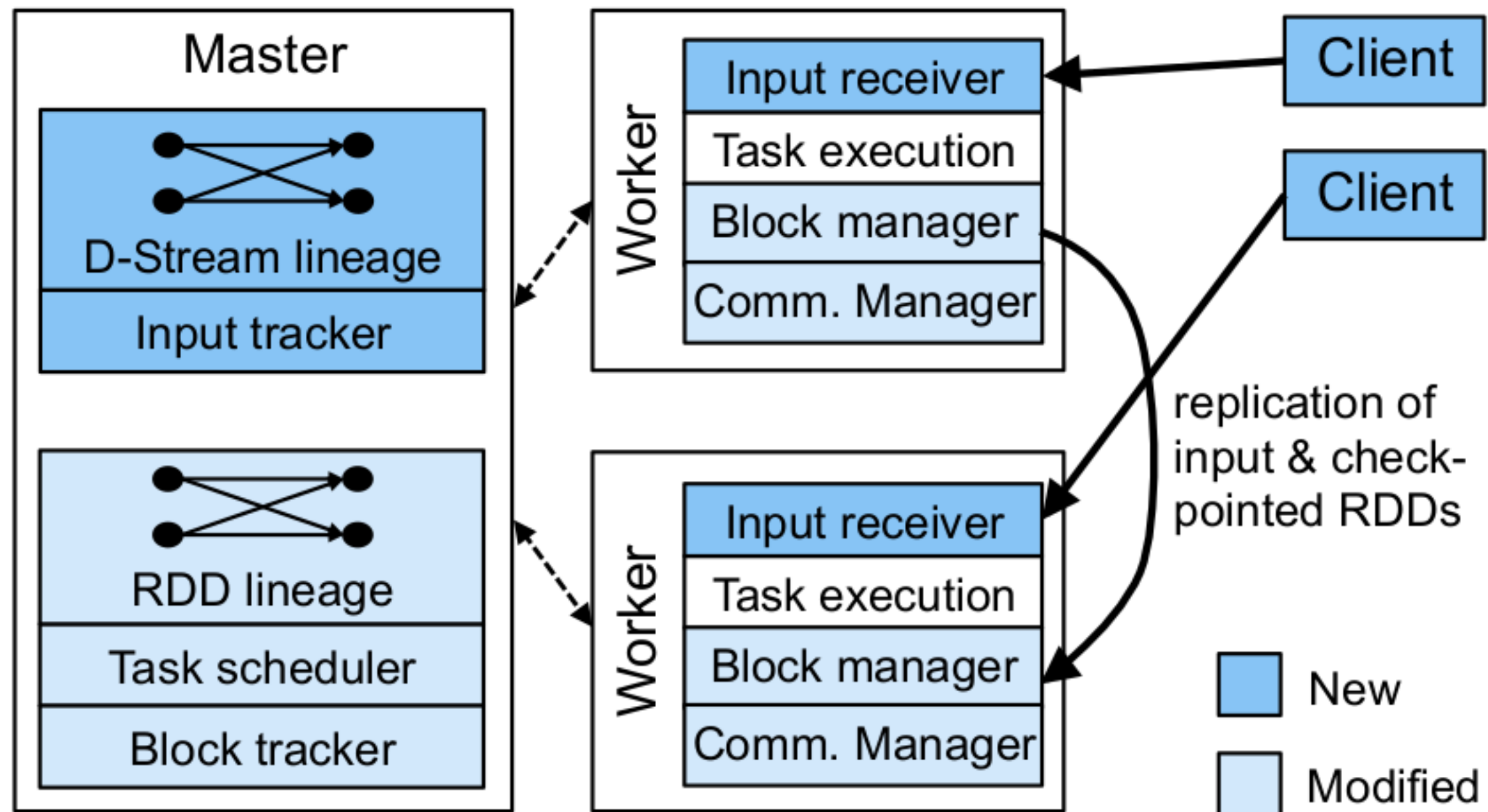


# Arquitetura - RDD

<b>Transformations</b>	<div><div><div><div><math>map(f : T \Rightarrow U)</math></div><div>:</div><div><math>RDD[T] \Rightarrow RDD[U]</math></div></div><div><div><math>filter(f : T \Rightarrow Bool)</math></div><div>:</div><div><math>RDD[T] \Rightarrow RDD[T]</math></div></div><div><div><math>flatMap(f : T \Rightarrow Seq[U])</math></div><div>:</div><div><math>RDD[T] \Rightarrow RDD[U]</math></div></div><div><div><math>sample(fraction : Float)</math></div><div>:</div><div><math>RDD[T] \Rightarrow RDD[T]</math> (Deterministic sampling)</div></div><div><div><math>groupByKey()</math></div><div>:</div><div><math>RDD[(K, V)] \Rightarrow RDD[(K, Seq[V])]</math></div></div><div><div><math>reduceByKey(f : (V, V) \Rightarrow V)</math></div><div>:</div><div><math>RDD[(K, V)] \Rightarrow RDD[(K, V)]</math></div></div><div><div><math>union()</math></div><div>:</div><div><math>(RDD[T], RDD[T]) \Rightarrow RDD[T]</math></div></div><div><div><math>join()</math></div><div>:</div><div><math>(RDD[(K, V)], RDD[(K, W)]) \Rightarrow RDD[(K, (V, W))]</math></div></div><div><div><math>cogroup()</math></div><div>:</div><div><math>(RDD[(K, V)], RDD[(K, W)]) \Rightarrow RDD[(K, (Seq[V], Seq[W]))]</math></div></div><div><div><math>crossProduct()</math></div><div>:</div><div><math>(RDD[T], RDD[U]) \Rightarrow RDD[(T, U)]</math></div></div><div><div><math>mapValues(f : V \Rightarrow W)</math></div><div>:</div><div><math>RDD[(K, V)] \Rightarrow RDD[(K, W)]</math> (Preserves partitioning)</div></div><div><div><math>sort(c : Comparator[K])</math></div><div>:</div><div><math>RDD[(K, V)] \Rightarrow RDD[(K, V)]</math></div></div><div><div><math>partitionBy(p : Partitioner[K])</math></div><div>:</div><div><math>RDD[(K, V)] \Rightarrow RDD[(K, V)]</math></div></div></div></div>
<b>Actions</b>	<div><div><div><div><math>count()</math></div><div>:</div><div><math>RDD[T] \Rightarrow Long</math></div></div><div><div><math>collect()</math></div><div>:</div><div><math>RDD[T] \Rightarrow Seq[T]</math></div></div><div><div><math>reduce(f : (T, T) \Rightarrow T)</math></div><div>:</div><div><math>RDD[T] \Rightarrow T</math></div></div><div><div><math>lookup(k : K)</math></div><div>:</div><div><math>RDD[(K, V)] \Rightarrow Seq[V]</math> (On hash/range partitioned RDDs)</div></div><div><div><math>save(path : String)</math></div><div>:</div><div>Outputs RDD to a storage system, <i>e.g.</i>, HDFS</div></div></div></div>



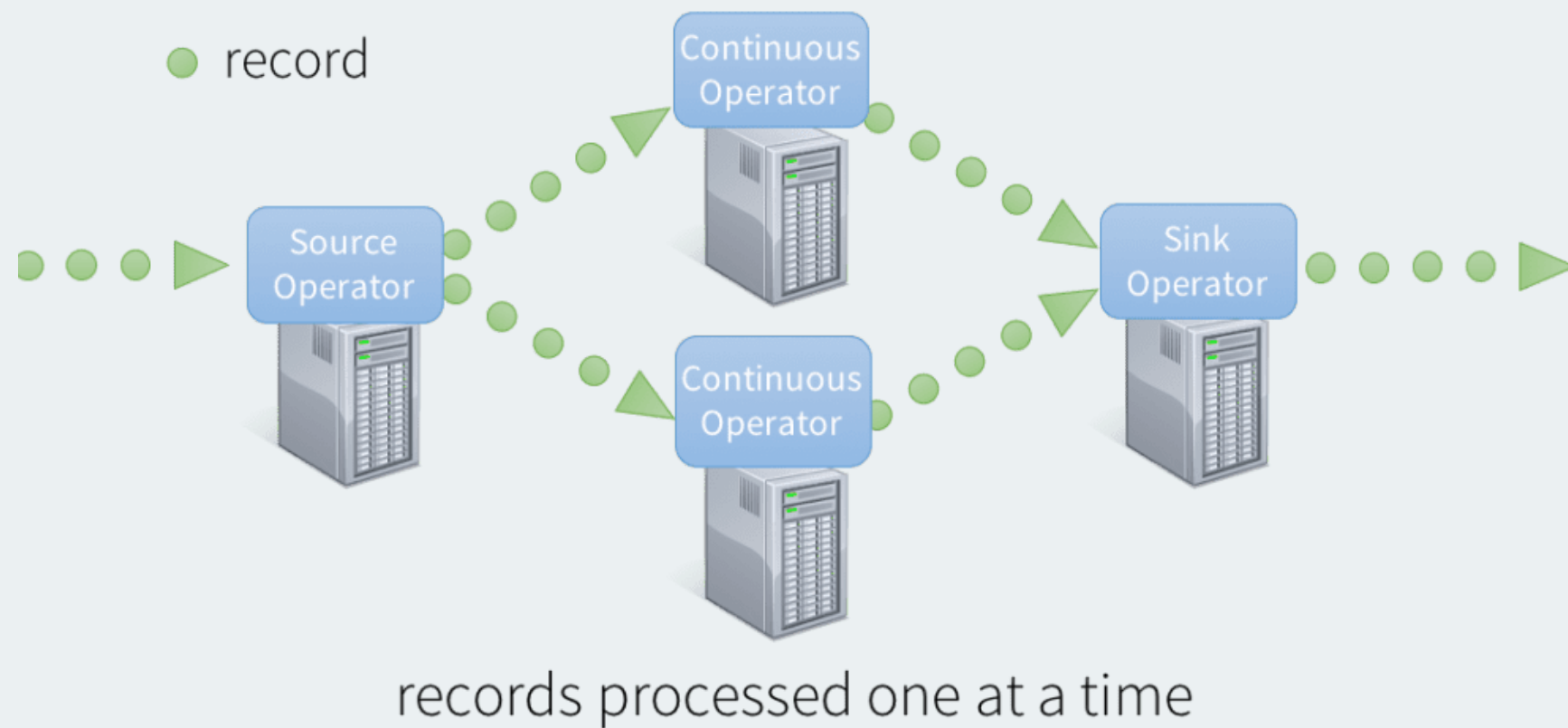
# Arquitetura



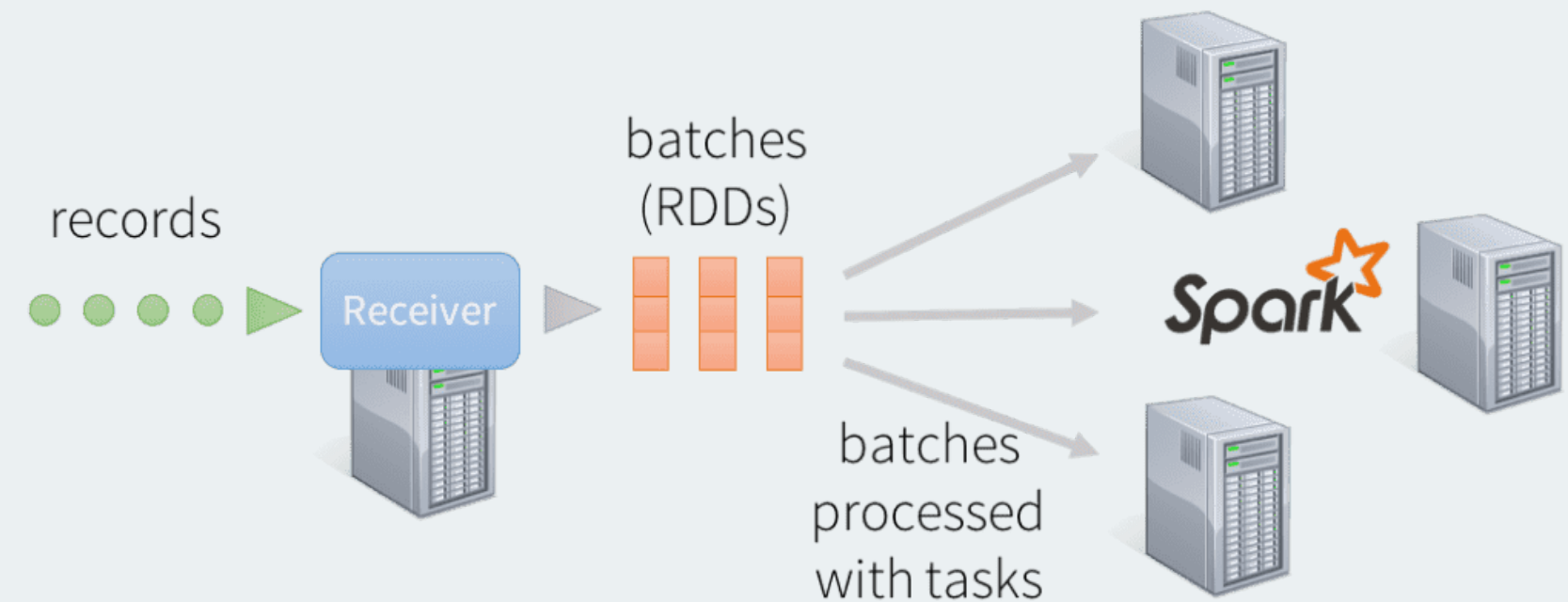
# ARQUITETURA

## Spark Streaming

Traditional stream processing systems  
*continuous operator model*



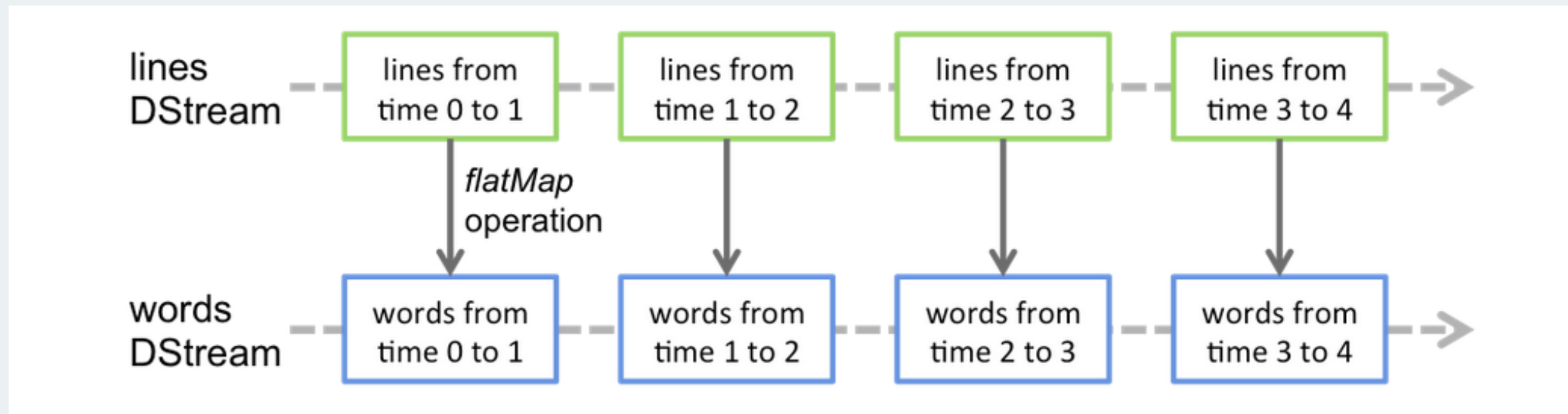
**Spark Streaming**  
*discretized stream processing*



records processed in batches with short tasks  
each batch is a RDD (partitioned dataset)

# ARQUITETURA

## Spark Streaming



- Discretized Stream ou **DStream** é a abstração básica fornecida pelo Spark Streaming.
- Representa um fluxo contínuo de dados. Internamente, um DStream é representado por uma série contínua de RDDs com a **mesma estrutura**, mas em **intervalos de tempo distintos**.

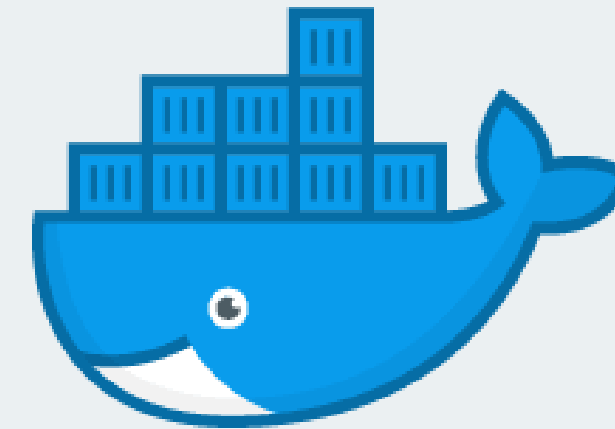
# Aplicação - Overview



# INSTALAÇÃO



Google Cloud



docker

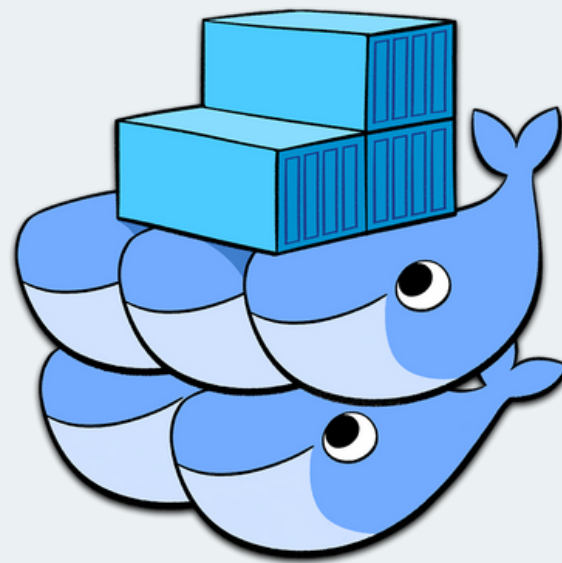


kubernetes



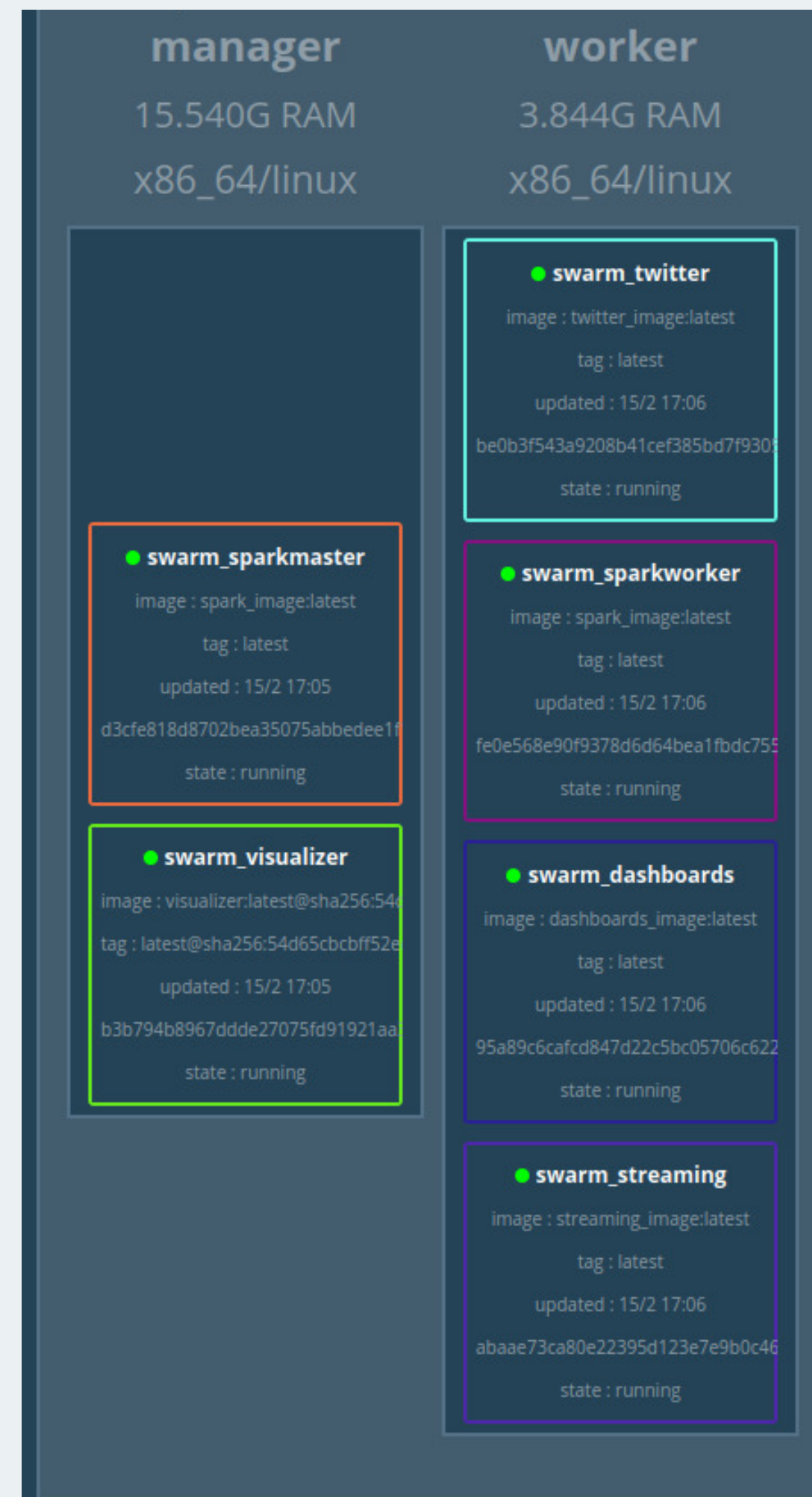
amazon  
EMR

# 1º TENTATIVA



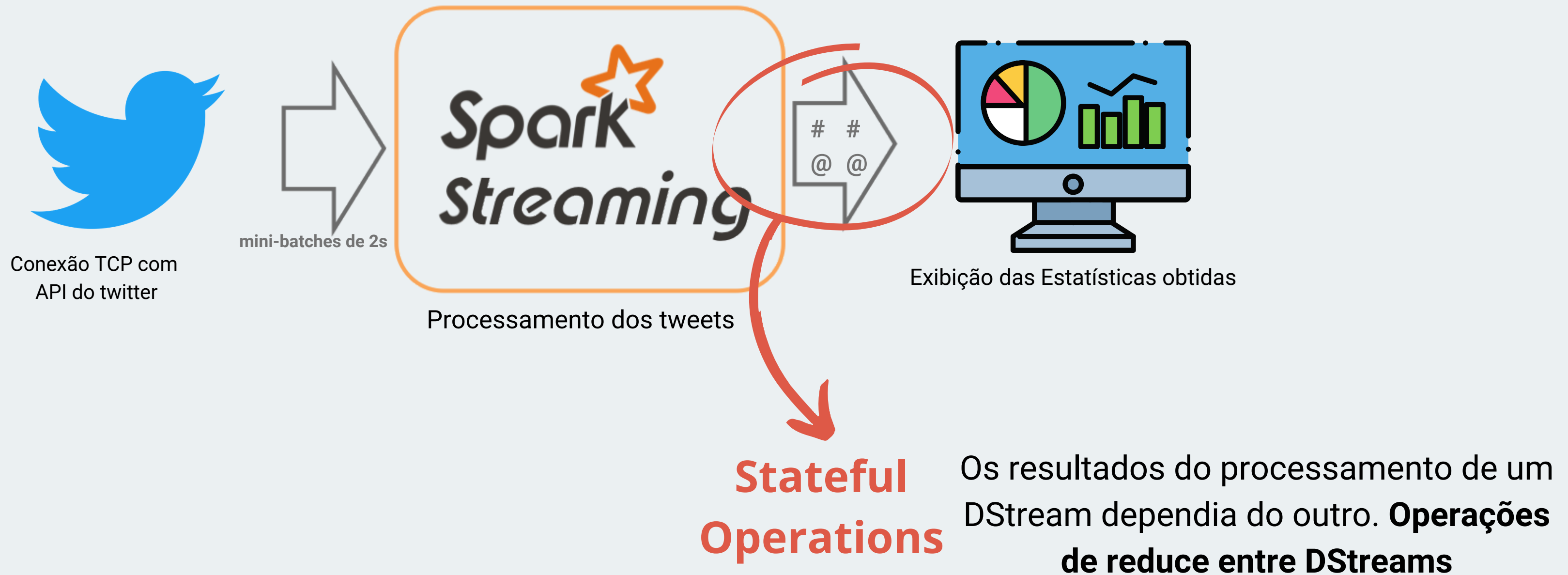
docker swarm

- A instalação do Cluster Spark utilizando VM's do Google Cloud foi um **sucesso**
- Porém, algo deu errado na nossa aplicação....



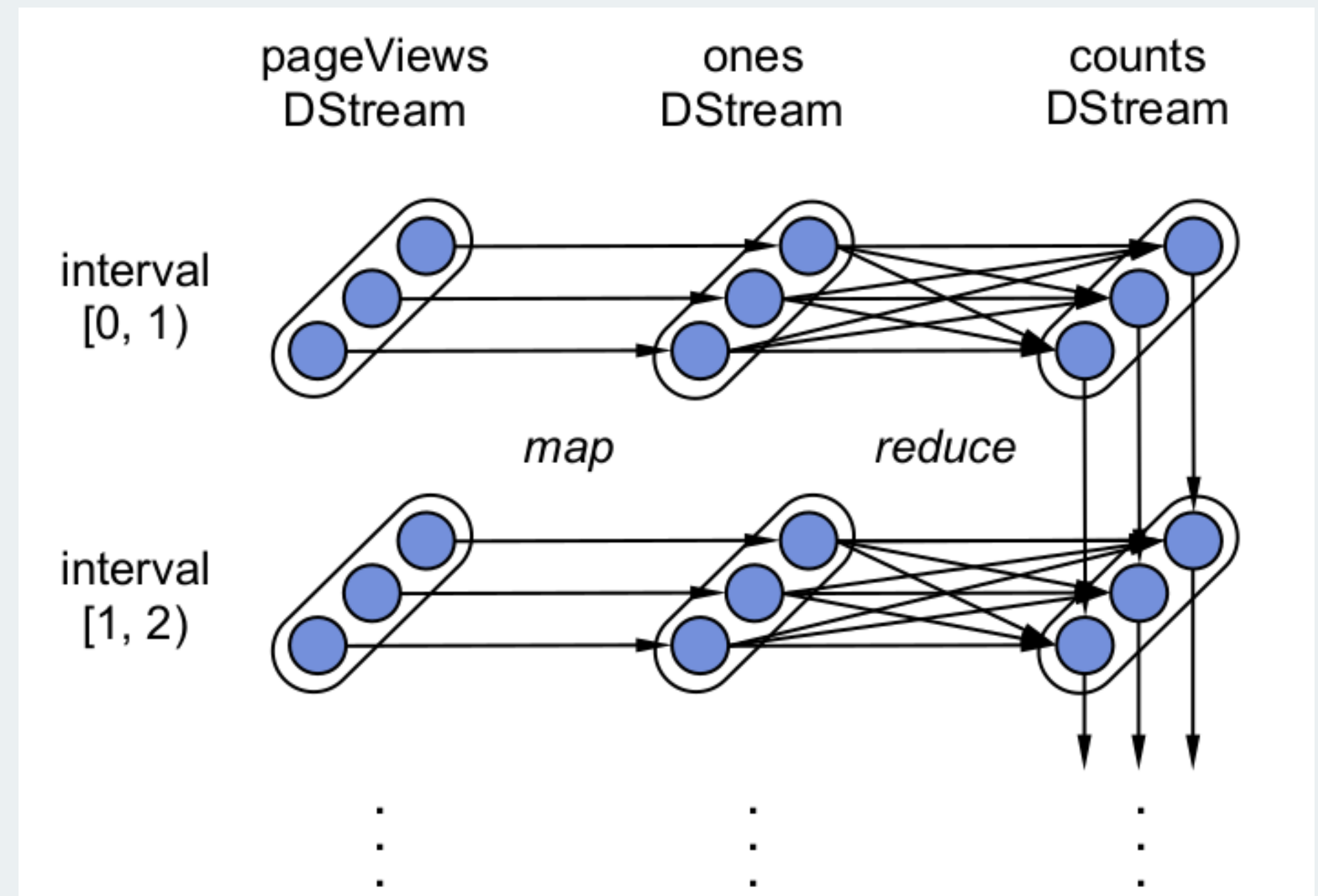


# Problemas no Deploy



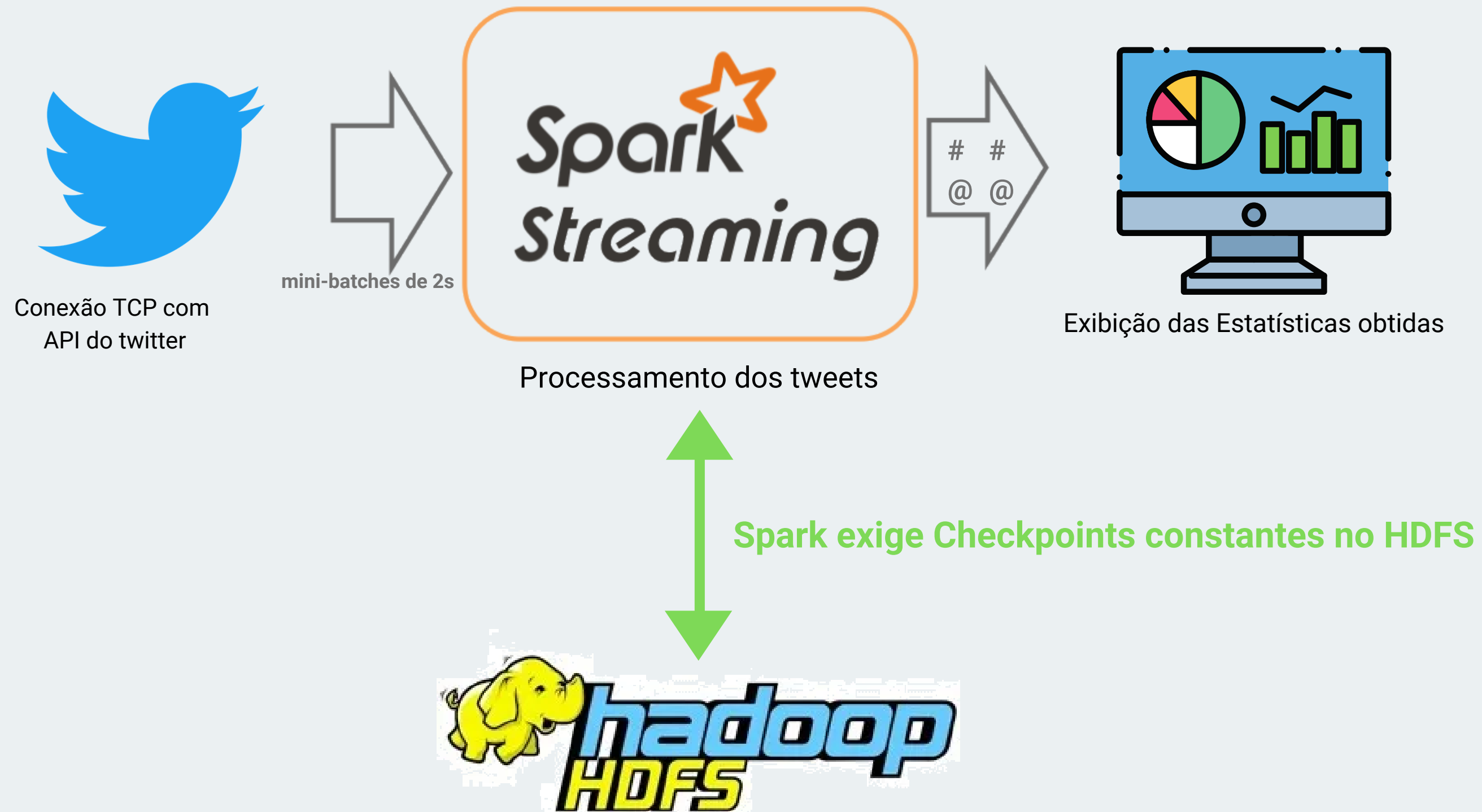
# CHECKPOINTING

- Isso é necessário em algumas transformações com estado que combinam dados de vários batches.
- Em tais transformações, os RDDs de um DStream dependem dos RDDs anteriores do mesmo DStream.

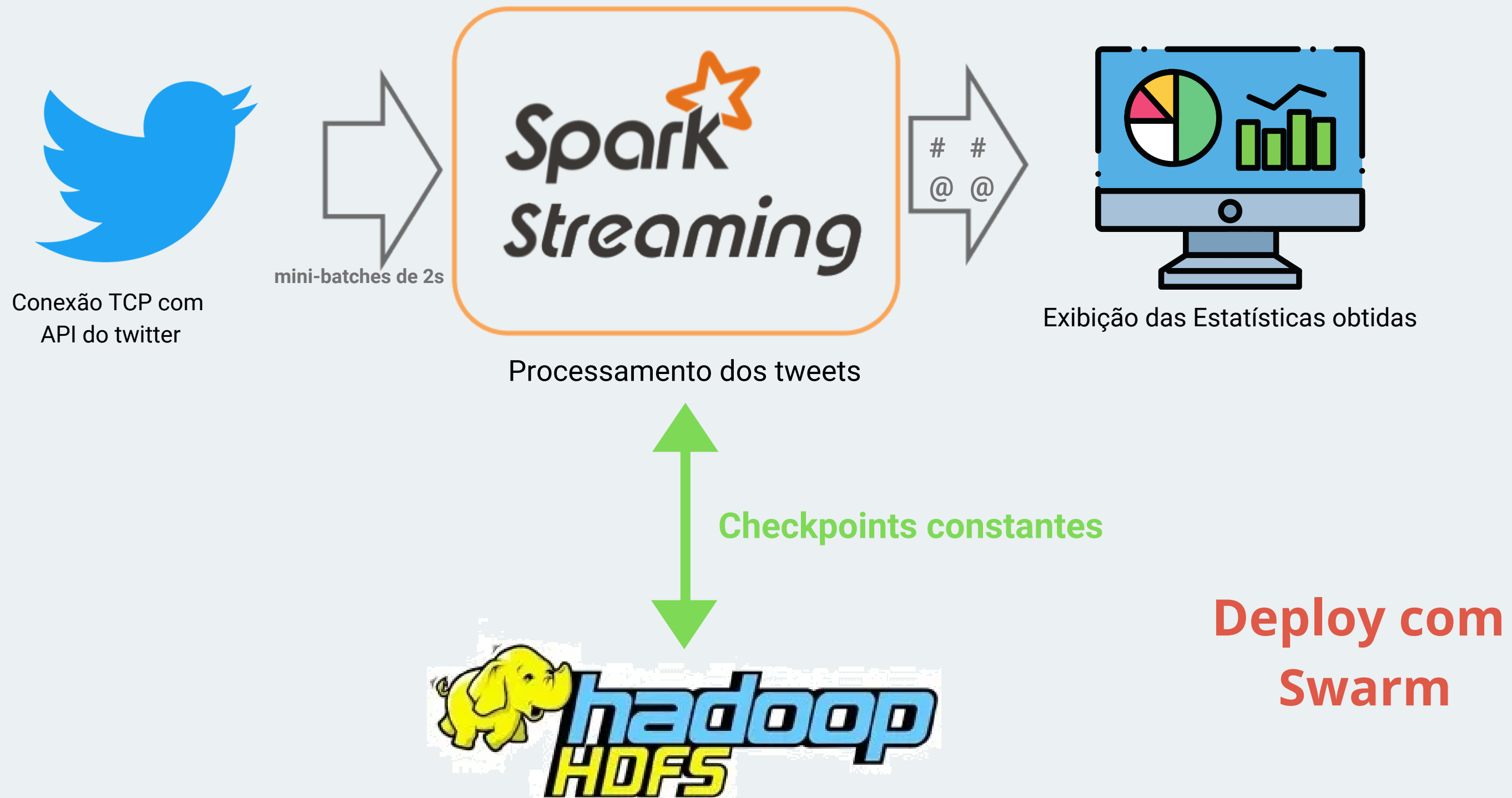




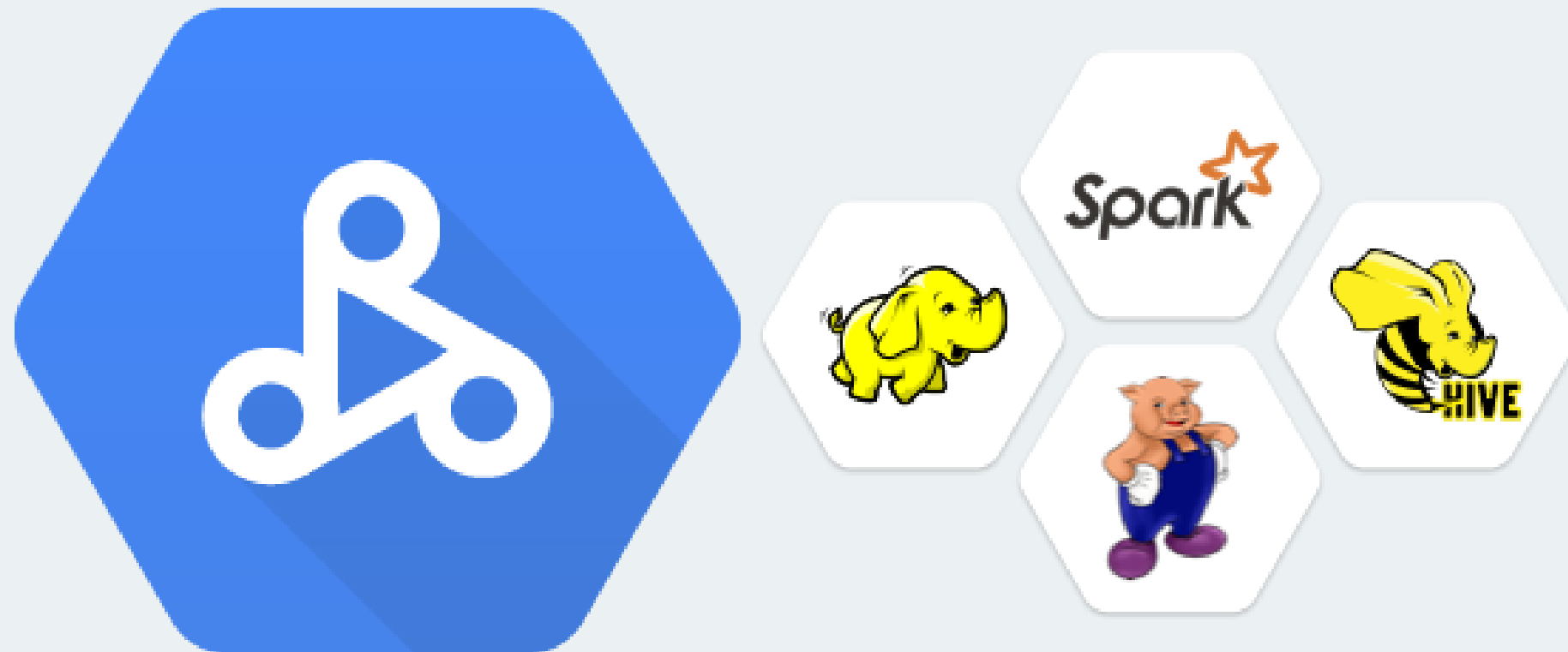
# Problemas no Deploy



# Problemas Avançados



## 2º Tentativa - Solução



- A alternativa foi abandonar os Containers e utilizar o cluster de máquinas do **Google Dataproc**
- O Dataproc já tem instalado no Cluster o Spark, juntamente com o Hadoop e o Yarn, o que era exatamente o que precisávamos

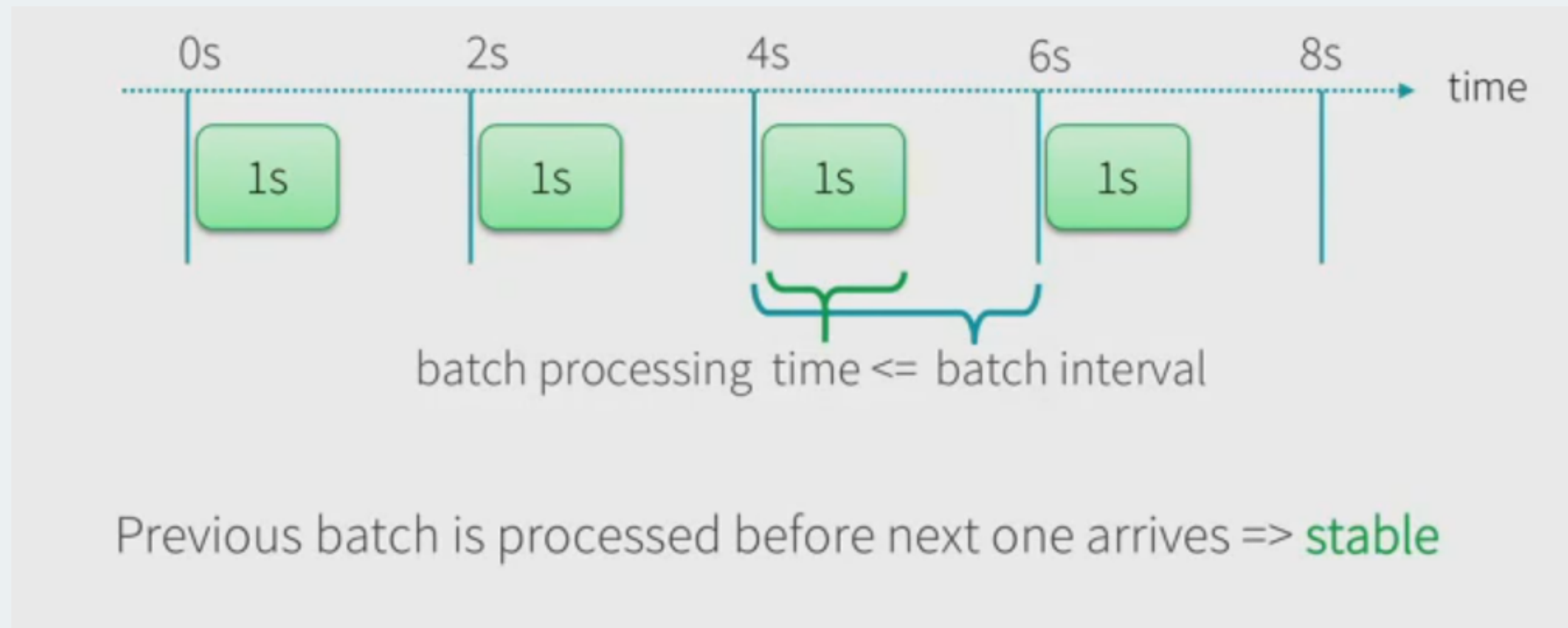
**Deploy com  
DataProc**



**obrigado**

# ADAPTATIVIDADE

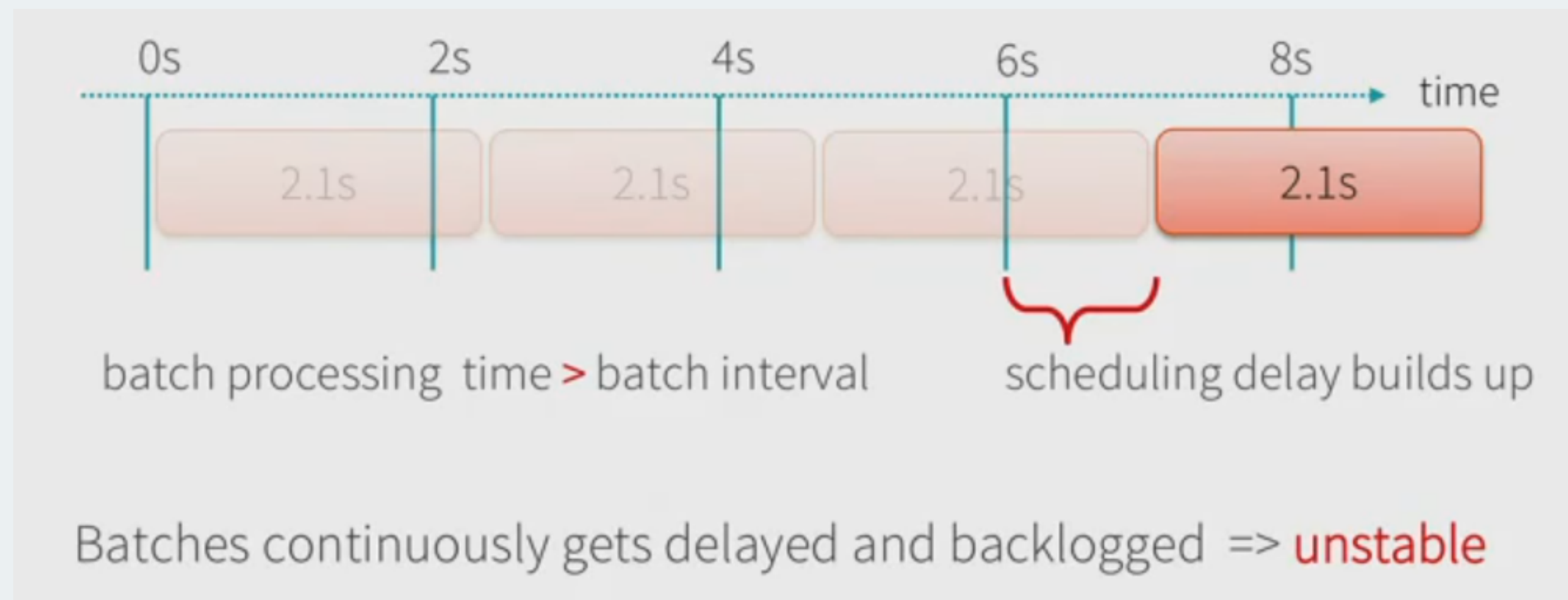
- A condição de estabilidade do Spark se passa por terminar o processamento do mini-batch antes do próximo chegar.



- Mas o que acontecerá se o sistema receber dados em uma taxa mais alta do que pode processar durante um pico de temporário???

# ADAPTATIVIDADE

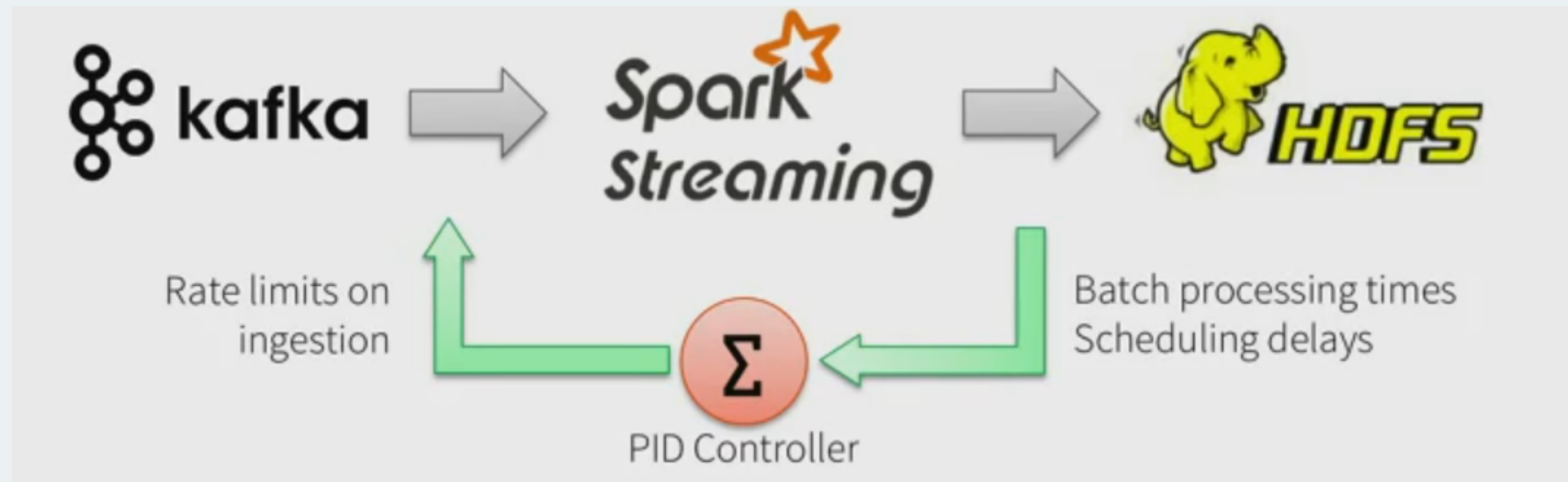
- *A condição de estabilidade do Spark se passa por terminar o processamento do mini-batch antes do próximo chegar.*



- **Mas o que acontecerá se o sistema receber dados em uma taxa mais alta do que pode processar durante um pico de temporário???**

# ADAPTATIVIDADE

## Backpressure



- O Spark Streaming estima o tempo de processamento do Batch e usa esta estimativa para limitar o tamanho do batch para os processamentos seguintes, evitando instabilidade no serviço.